PIP Chill Documentation

Release 0.1.3

Ricardo Bánffy

Jan 22, 2018

Contents

Contents:

CHAPTER 1

PIP Chill - Make requirements with only the packages you need

Like *pip freeze* but lists only the packages that are not dependencies of installed packages.

- Free software: GNU General Public License v3
- Documentation: https://pip-chill.readthedocs.io.

1.1 Features

Generates a requirements file without any packages that depend on other packages in the file.

1.2 Usage

Suppose you have installed in your virtualenv a couple packages. When you run *pip freeze*, you'll get a list of all packages installed, with all dependencies. If one of the packages you installed ceases to depend on an already installed package, you have to manually remove it from the list. The list also makes no distinction about the packages you actually care about and packages your packages care about, making the requirements file bloated and, ultimately, inaccurate.

On your terminal, run:

```
$ pip-chill
asciitree==0.3.1
autopep8==1.2.4
beautifulsoup4==4.4.0
bleach==1.4.1
cookiecutter==1.4.0
coverage==3.7.1
django-argonauts==1.0.1
...
```

Or, if you want it without version numbers:

```
$ pip-chill --no-version
asciitree
autopep8
beautifulsoup4
bleach
cookiecutter
coverage
django-argonauts
...
```

Or, if you want to list package dependencies too:

```
$ pip-chill -v
asciitree==0.3.1
autopep8==1.2.4
beautifulsoup4==4.4.0
bleach==1.4.1
cookiecutter==1.4.0
coverage==3.7.1
django-argonauts==1.0.1
# arrow==0.10.0 # Installed as dependency for jinja2-time
# binaryornot==0.4.4 # Installed as dependency for cookiecutter
# chardet==3.0.4 # Installed as dependency for binaryornot
# click==6.7 # Installed as dependency for cookiecutter
# django==1.11.5 # Installed as dependency for django-argonauts
# future==0.16.0 # Installed as dependency for cookiecutter
# html5lib==0.999999999 # Installed as dependency for bleach
# jinja2==2.9.6 # Installed as dependency for jinja2-time, cookiecutter
# jinja2-time==0.2.0 # Installed as dependency for cookiecutter
# markupsafe==1.0 # Installed as dependency for jinja2
# pep8==1.7.0 # Installed as dependency for autopep8
# poyo==0.4.1 # Installed as dependency for cookiecutter
# python-dateutil==2.6.1 # Installed as dependency for arrow
# pytz==2017.2 # Installed as dependency for django
# six==1.11.0 # Installed as dependency for python-dateutil, html5lib, bleach
# webencodings==0.5.1 # Installed as dependency for html5lib
# whichcraft==0.4.1 # Installed as dependency for cookiecutter
...
```

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install PIP Chill, run this command in your terminal:

```
$ pip install pip_chill
```

This is the preferred method to install PIP Chill, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for PIP Chill can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/rbanffy/pip_chill
```

Or download the tarball:

```
$ curl  -OL https://github.com/rbanffy/pip_chill/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

To use PIP Chill in a project:

```
import pip_chill
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/rbanffy/pip_chill/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

PIP Chill could always use more documentation, whether as part of the official PIP Chill docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/rbanffy/pip_chill/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *pip_chill* for local development.

1. Fork the *pip_chill* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pip_chill.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pip_chill
$ cd pip_chill/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pip_chill tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/rbanffy/pip_chill/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_pip_chill
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search